Seminar 1 - Kombinatorik

▷ Ziehen mit Wiederholung (mit Zurücklegen): Auswahl (Ziehen) von Elementen bei der das gewählte (gezogene) Element sofort wieder der Gesamtmenge zugefügt wird.

▷ Ziehen ohne Wiederholung (ohne Zurücklegen): Auswahl (Ziehen) von Elementen bei der das gewählte (gezogene) Element *nicht* wieder der Gesamtmenge zugefügt wird.

Definition Permutation: Jede mögliche Anordnung von n Elementen, in der alle Elemente verwendet werden, heißt Permutation dieser Elemente.

Permutation ohne Wiederholung ($n \in \mathbb{N}$):
Von n paarweise verschiedenen Elementen gibt es $n!$ Permutationen (also Anordnungsmöglichkeiten).
Konvention: $0!=1$
Beispiel: 1) Wie viele Möglichkeiten gibt es, 5 Personen in eine Reihe zu setzen?
Lösung: $\rightarrow$ Permutation ohne Wiederholung: $5! = 120$

```
import itertools
from itertools import permutations
for p in permutations("DEFG"):
    print("".join(p))
```

Permutation mit Wiederholung ($n, r \in \mathbb{N}^*, r \leq n$):
Gegeben seien n Elemente in r Gruppen, wobei sich die Elemente einer Gruppe nicht unterscheiden, die Elemente verschiedener Gruppen aber verschieden sind. Die Anzahl der Elemente in den Gruppen sei $i_1, i_2, \ldots, i_r$ ($i_1 + i_2 + \cdots + i_r = n$). Zu diesen Elementen gibt es

$$\frac{n!}{i_1! \cdot i_2! \cdot \ldots \cdot i_r!}.$$

Permutationen (Anordnungsmöglichkeiten).
Beispiel: 1) Wie viele Möglichkeiten gibt es aus den Zahlen 1 bis 4 jeweils 7-stellige Zahlen zu bilden, in denen 2-mal die 1, 3-mal die 2 und je einmal die 3 bzw. die 4 vorkommen?
Lšung: $\rightarrow$ Permutation mit Wiederholung:

$$\frac{7!}{2!3!1!1!} = 420.$$

2) In Python: alle (verschiedenen) Permutationen von `AABB` sind: `AABB`, `ABAB`, `ABBA`, `BAAB`, `BABA`, `BBAA` (es sind Permutationen mit Wiederholung).

```
import more_itertools
from more_itertools import distinct_permutations
for p in distinct_permutations("AABB"):
   print("".join(p))
```

Definition Kombination:
Jede mögliche Anordnung ohne Berücksichtigung der Reihenfolge aus n Elementen je k heißt Kombination dieser Elemente (Kombination von n Elementen zur k-ten Klasse).

Kombination ohne Wiederholung ($k \in \mathbb{N}, n \in \mathbb{N}^*, k \leq n$):
Aus n verschiedenen Elementen können k Stück ohne Berücksichtigung der Reihenfolge und ohne Zurücklegen auf

$$C_n^k = \frac{n!}{k!(n-k)!}.$$

verschiedene Arten ausgewählt werden.
Beispiel: 1) Wie viele Möglichkeiten gibt es aus 10 Schülern 4 auszuwählen?
Lösung: $\rightarrow$ Kombination ohne Wiederholung: $C_{10}^{4} = 210$.
2) Kombinationen in Python:

```
import math
from math import comb
import itertools
from itertools import combinations
print("alle Kombinationenen aus DEFGH je 3")
for p in combinations("DEFGH",3):
    print("".join(p))
print("Wert Kombination C_5^3:", comb(5,3))
```

Kombination mit Wiederholung (für $n \in \mathbb{N}^*$, $k \in \mathbb{N}$):
Aus n verschiedenen Elementen wird k-mal hintereinander eines ausgewählt und vor dem nächsten Zug wieder zurückgelegt. Dann gibt es ohne Berücksichtigung der Reihenfolge insgesamt

$$C_{n+k-1}^{k} = \frac{(n+k-1)!}{k!(n-1)!}$$

verschiedene Auswahlmöglichkeiten.
Beispiel: 1) Wie viele Möglichkeiten gibt es 5 (nicht unterscheidbare) Äpfel auf 3 Kinder (A,B,C) zu verteilen? (mehrere oder keine Äpfel können an ein Kind verteilt werden) Beispiele von Aufteilungen (0 ist "Apfel"):

A	B	C
00		000
0	0	000
000	00	
		00000

Erste Methode: Wir können alle Ergebnisse durch Ziffernfolgen kennzeichnen, wobei 0 für einen "Apfel" und 1 für eine "Trennwand" zwischen den Kindern steht:
a) 0011000 b) 0101000 c) 0001001 d)1100000
Reformulierung: Wie viele Möglichkeiten gibt es, 5 Nullen und 2 Einsen auf 7 Positionen zu verteilen? Antwort: $C_7^5 = C_7^2 = \frac{7!}{5!2!} = 21$.
Zweite Methode: Kombinationen mit Wiederholung $\rightarrow$ Kombination mit Wiederholung: n = 3, k = 5; aus der Menge mit 3 Kindern, wählt man 5-mal, mit Wiederholung; bei a) A,A,C,C,C, bei b) A,B,C,C,C, bei c) A,A,A,B,B, bei d) C,C,C,C,C
2) Kombination mit Wiederholung in Python:

```
import itertools
from itertools import combinations_with_replacement
for p in combinations_with_replacement("ABC",5):
  print("".join(p))
```

Definition Variation:
Jede mögliche Anordnung (mit Berücksichtigung der Reihenfolge) aus je k Elementen von n Elementen heißt Variation dieser Elemente (Variation von n Elementen zur k-ten Klasse).
Variation ohne Wiederholung ($k \in \mathbb{N}, n \in \mathbb{N}^*, k \leq n$):
Aus n verschiedenen Elementen können k Stück mit Berücksichtigung der Reihenfolge und ohne Zurücklegen auf

$$V_n^k = n \cdot (n-1) \cdot \ldots \cdot (n-k+1) = \frac{n!}{(n-k)!}.$$

verschiedene Arten ausgewählt werden.
Beispiel: Wie viele Kodes mit 3 verschiedenen Ziffern aus der Menge $\{1, 2, 3, 4\}$ kann man bilden?
Lösung: $V_4^3 = \frac{4!}{(4-3)!} = 24$.
2) Alle Variationen von `ABC` je 2 in Python:

```
import itertools
from itertools import permutations
for p in permutations("ABC",2):
    print("".join(p))   # alle Variationen von ABC je 2
```

Variation mit Wiederholung (für $k, n \in \mathbb{N}^*$):
Aus n verschiedenen Elementen werde k-mal hintereinander eines ausgewählt und vor dem nächsten Zug wieder zur Grundmenge zurückgelegt. Dann gibt es unter Berücksichtigung der Reihenfolge insgesamt n^k verschiedene Auswahlmöglichkeiten.
Bemerkung: Die **Anzahl der Funktionen** von der Menge A mit k Elementen zu der Menge B mit n Elementen $(k, n \in \mathbb{N}^*)$ ist n^k und entspricht der Variation mit Wiederholung.

Beispiel: Wie viele Kodes mit 4 Ziffern kann man bilden?
Lösung: 10^4.
Beispiel: 1) Wie viele Möglichkeiten gibt es eine Orange, einen Apfel, eine Banane an 4 Kinder aufzuteilen? (die Früchte werden nicht geschnitten; ein Kind kann mehrere Früchte erhalten).
Lösung: die Anzahl der Funktionen $f : \{\text{“Orange”}, \text{“Apfel”}, \text{“Banane”}\} \to \{K1, K2, K3, K4\}$ ist 4^3.
2) Alle Variationen von `ABC` je 2 mit Wiederholung in Python: `AA, AB, BA, AC, CA, BB, BC, CB, CC`.

```
import itertools
from itertools import combinations_with_replacement
import more_itertools
from more_itertools import distinct_permutations
c=0
print("Alle Variationen von ABC je 2 mit Wiederholung:")
for p in combinations_with_replacement("ABC",2):
  for t in distinct_permutations(p):
    print("".join(t))
    c=c+1
print("Anzahl Variationen von ABC je 2 mit Wiederholung:",c)
```

Klassische Definition der Wahrscheinlichkeit:
Wir betrachten ein Experiment welches endlich viele, gleichwahrscheinliche Ergebnisse hat. Die Wahrscheinlichkeit, dass das Ereignis A eintretet ist

$$P(A) = \frac{\text{Anzahl der günstigen Fälle für das Eintreten von } A}{\text{Anzahl aller möglichen Fälle innerhalb des Experiments}}.$$

Beispiel: Welches ist die Wahrscheinlichkeit bei einem Lottoschein “6 aus 49” alle 6 Zahlen richtig auszufüllen? (die Reihenfolge der Zahlen wird nicht in Betracht gezogen)
Lösung: $\dfrac{1}{C_{49}^6}$.